

**Dynamic job assignment  
heuristics for multi-server  
batch operations : a cost  
based approach**

D.J. van der Zee, A. van Harten  
and P.C. Schuur

R 19

BETA publicatie

Eindhoven

Keywords

BETA-Research Programme

Reprinted from:

R 19 (reprint)

December 1998

Unit management

International Journal  
of Production

Research, vol. 35,

no. 11(1997):3063-

3093

## Dynamic job assignment heuristics for multi-server batch operations – A cost based approach

D. J. VAN DER ZEE†\*, A. VAN HARTEN† and P. C. SCHUUR†

In many industries production facilities are used which process products in a batch-wise manner. Guided by research in the aircraft industry, where the process of hardening synthetic aircraft parts was studied, we propose a new control strategy for these types of systems. Given the availability of information on a few near future arrivals the strategy decides on when to schedule a job in order to minimize logistical costs. The fact that different cost structures can be incorporated, makes it a valuable tool for use in practical situations in business. The potential of the new strategy is demonstrated by an extensive series of simulation experiments, in which its response for various system configurations was tested in comparison with existing heuristics.

### 1. Introduction

This work arose from a planning problem in the aircraft industry (Hodes *et al.* 1992), concerning the development of a control system for the production of hardened synthetic parts. Because of the highly competitive nature of the aircraft industry, lead time reductions and improvements in the service level are of vital importance. The proposed new oven control system contributes in that direction.

Synthetic aircraft parts are hardened in an oven in a batch-wise manner. Since hardening requires specific temperatures, pressures and processing times, several product groups for these parts are distinguished. A batch consists of parts from one product group. Batch sizes are limited by the physical size of the oven and by a process constraint, which determines a maximum fill rate for the oven. Processing times (including preparation times) are considered to be constant and independent of the number of parts in a batch. Once processing has started, no interruption is allowed, i.e. no addition or extraction of parts is possible during the production process. Given these characteristics, a control strategy is used to schedule a number of ovens in such a way that an optimal trade-off between logistical costs and the service level is reached. The possibilities for scheduling the ovens are depending on the availability of information on (predicted) future arrivals of parts of different types.

The described system is known in literature as a *bulk queueing system*. Bulk queueing systems are characterized by the fact that customers arrive in groups and/or are served in groups. Apart from ovens, many other examples of such systems can be given. Deb and Serfozo (1973) and Bagchi and Templeton (1972) mention elevators, charter airline flights, shuttle buses, mass transportation systems,

---

Revision received November 1996.

†Department of Management Science and Logistics, Faculty of Technology and Management, University of Twente, P.O. Box 217, 7500 AE Enschede, The Netherlands.

\*To whom correspondence should be addressed.

transshipment of mail, restaurants and the production, inventory control and shipment of commercial products. Systems which are quite similar to the system described here can be found in the manufacturing of integrated circuits (Uzsoy *et al.* 1992, Fowler *et al.* 1992). Here, the processing steps, diffusion and oxidation, require the placement of wafers in furnace tubes.

Control strategies for bulk queueing systems can be classified according to the amount of information which is supposed to be available about future arrivals of customers. Three typical situations can be distinguished:

- No information available;
- Full knowledge of future arrivals;
- A few near-future arrivals are known or predicted.

The first category refers to control strategies which are based upon the information about the current situation only. An important example of such a strategy for the single machine case, the Minimum Batch Size (MBS) rule, was introduced by Neuts (1967). According to this strategy, a batch starts service as soon as at least a certain fixed number of customers is present. Using a dynamic programming approach, Deb and Serfozo (1973) showed how this critical load should be chosen in order to minimize the expected discounted cost over an infinite horizon. If the cost of serving is set to zero and the cost of waiting is linear, minimizing the expected averaged cost is equivalent to minimizing the average waiting time (Fowler *et al.* 1992).

Bulk queueing systems where full knowledge is available about future arrivals are studied in the field of deterministic machine scheduling. An overview of such strategies is given by Uzsoy *et al.* (1994), who discuss planning and scheduling models applicable to the semiconductor industry. The relevance of models like these is quite limited, because, in practice, only little information on future arrivals is available.

Let us now discuss control strategies of the third type. This type of control strategies, the so-called 'look-ahead strategies', has been studied only quite recently. Glassey and Weng (1991) address this subject for semiconductor manufacturing. They discuss the practical usability of a dynamic programming approach to find a sequence of loading times of given lots, in such a way that the total delay is minimized. They argue that this approach fails for reasons of computational feasibility, and the availability and quality of data on future arrivals. Therefore they present a 'dynamic batching heuristic' (DBH). The heuristic decides when to start a new production cycle thereby aiming for a minimal average waiting time. The planning horizon in DBH is just one processing time. DBH proves to perform better than MBS, based upon the knowledge of just a few future arrivals. The DBH-heuristic was modified by Fowler *et al.* (1992) and was extended to the multi-product single server bulk queueing system. Their 'next arrival control heuristic' (NACH) considers only the next arrival. Weng and Leachman (1993) derive a scheduling rule which is similar to the discrete time dynamic demand lot-sizing heuristic proposed by Silver and Meal (1973). Their heuristic, the 'minimum cost rate' heuristic (MCR), is found to perform slightly better than DBH and NACH for moderate traffic intensity for the single product single machine case. For the multi-product single machine case Weng and Leachman state MCR to be superior to NACH, although the heuristics are only compared for one specific setting for product specifications. The robustness of MCR was improved by Robinson *et al.* (1995). They incorporated a rolling horizon into the cost rate function used in MCR. The resulting Rolling Horizon

Cost Rate heuristic (RHCR) was further extended to address a batch-serial system, where batch processing is followed by a serial step.

Although the look-ahead strategies mentioned above give good results in comparison with heuristics which only use local information, like MBS, they are only meant to optimize average flow time while neglecting other logistical costs. In this paper a new heuristic, the Dynamic Job Assignment Heuristic (DJAH) will be discussed (van der Zee 1997). It is capable of dealing with situations where *one or multiple types of products* have to be processed by a number of *identical machines*. In addition to the costs associated with waiting, also setup costs for a machine are taken into account. Here, setup costs refer to e.g. the manpower needed for loading an oven, the use of vehicles for transportation of parts and the consumed energy. If setup costs are absent and the waiting costs are linear, DJAH focuses on a minimal average waiting time. This heuristic can easily be adapted to include other cost functions in order to reflect company dependent cost structures, such as penalty costs for late deliveries or maintenance costs.

The paper runs as follows. In §2 the above look-ahead strategies, including DJAH, are introduced in detail. In subsections 2.1–2.4 the existing heuristics are surveyed. Next, in subsection 2.5 we propose the new DJAH heuristic. To show the potential of DJAH it was extensively tested by simulation. Its response was analysed for various system configurations, which reflect different settings, e.g. processing times, machine capacities and number of machines. Special attention is paid to the robustness of the heuristic in case of forecasting errors or missing data. For each of the settings a comparison with the other heuristics mentioned above has been carried out. In §3 we describe the underlying research methodology for these simulations. The results of these simulations are discussed in §4. Finally, in §5 we summarize the conclusions and indicate possibilities for further research.

## 2. Look-ahead strategies

In order to gain a clear understanding of the meaning of look-ahead strategies for the control of batch operations, we now formulate some additional assumptions with regard to the characteristics of these systems. Given these characteristics it is possible to classify control strategies according to their applicability for different system configurations. This classification will be used as a framework for the discussion of the developed control strategies. To support our discussion Fig. 1 will be used, which graphically depicts the process of hardening synthetic aircraft parts.

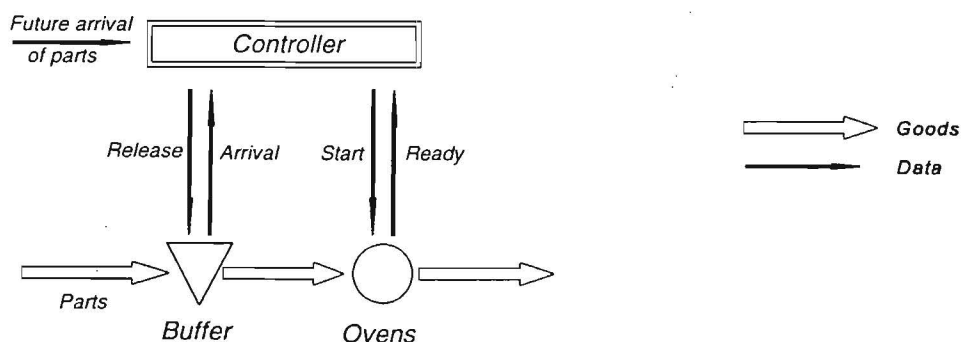


Figure 1. Hardening synthetic aircraft parts.



The manufacturing system for hardening synthetic parts is made up of three subsystems: a buffer, a number of ovens and a controller. The buffer is meant to decouple the system from any preceding manufacturing or transportation system by storing parts of different types until they are loaded into the oven. The capacity of this buffer is supposed to be unlimited. We consider the case that parts arrive individually. Compound arrival processes can be dealt with as a straightforward extension.

Parts are hardened batch-wise in one of the available ovens. Therefore the parts have to be transported from the buffer to the oven and a number of setup activities have to take place. The latter activities are assumed to be sequence-independent, i.e., cost/time depends only on the next batch to be processed. The duration of the setup activities is included in the processing time, which has a fixed length that depends on the type of product only. It is not allowed to interrupt processing, because this would make the respective parts worthless for further use. The number of parts to be manufactured is limited by the physical size of the oven and a process constraint, which determines a maximum fill rate for the oven. Given these constraints and the fact that we study identical ovens only, the maximum processing capacity for an oven depends only on the physical characteristics of a part type.

To control the job shop a decision maker is made responsible for assigning jobs to the ovens. The decision maker can be a human planner or may be integrated in an automatic control system. Control criteria are the service level as it is determined by the lead time (uncertainty) and the logistical costs associated with storage of parts and setup activities. The decision maker bases his decision upon the information received from other departments or external suppliers about the future arrivals of parts, and messages which report the actual arrivals of parts and the completion of oven cycles. The information about future arrivals covers a quite limited horizon. Moreover, data can be incomplete and/or subject to forecasting errors. Here we consider decision rules based on a rolling or jumping horizon approach. Two possible situations (decision moments) arise in which a decision has to be made:

- (1) An oven becomes idle and the number of lots in queue is greater than zero;
- (2) The arrival of a lot while an oven is idle.

Any event of this sort triggers a decision, where the decision options are:

- (1) Postponement to the next decision moment;
- (2) Acceptance of one (or more) job(s) for which the oven number(s) and product type(s) are specified, either immediately or at a specified future moment.

For example, the DBH and MCR rules do not postpone a decision, if more favourable conditions are foreseen in future. Instead of postponing the decision they determine the number of arrivals to wait before loading the oven. In this way they allow for jumping ahead, while skipping some intermediary decision moments.

Now it is possible to classify the control strategies, including the newly developed DJAH. As criteria for classification we apply the number of product types, the number of available machines, the objective for optimization and the look-ahead horizon (see Table 1).

The table starts with the MBS-rule for the single product single machine case. An adapted version of the MBS-rule, which covers the multiple products multiple machines case, **MBSX**, is introduced in subsection 2.1. Optimization is directed towards the minimization of logistical costs and/or the minimization of average

Standard heuristic	(No. of machines, no. of products)				Objective	Information horizon	
	(1, 1)	(1, N)	(M, 1)	(M, N)		Jumping	Length
MBS	x		x		c, f	no	0
MBSX	x	x	x	x	f	no	0
DBH	x	x	x	x	f	yes	kA, T
NACH	x	x			f	no	1A
MCR	x	x			c, f	yes	kA, T
RHCR	x	x			c, f	no	kA, T
DJAH	x	x	x	x	c, f	no	1A, 1M
MBS	Minimum Batch Size rule				c	Minimal cost	
MBSX	Adapted Minimum Batch Size rule				f	Minimal average flow time	
DBH	Dynamic Batching Heuristic				kA	k arrivals ahead	
NACH	Next Arrival Control Heuristic				1M	The moment at which the first machine is available	
MCR	Minimum Cost Rate Heuristic						
RHCR	Rolling Horizon Cost Rate Heuristic				T	Processing time	
DJAH	Dynamic Job Assignment Heuristic						

Table 1. A classification of control strategies.

flow time in the long term. It should be noted that, given the fixed processing times, minimization of average flow time is in fact equivalent to minimization of average waiting time. Typically, the heuristics try to reach these goals by adopting:

- (1) A reduced criterion for optimization; and
- (2) Evaluating a limited number of decision options.

The choice of options is related to the look-ahead horizon, to the presence of full loads and sometimes to other situational aspects. Any decision rule we consider uses a finite horizon, which is either fixed or arrival-dependent. In the first case, information on arrivals up to  $T$  time units ahead is known at any moment. It is natural to relate  $T$  to the maximal processing time of a job (e.g. DBH). In the second case, information on the next arrival or the next  $k$  arrivals for each product type is given (e.g. NACH). We refer to these possibilities as '1A' and 'kA' respectively. Of course a superposition of both types is also possible, which is the case for MCR and DJAH. The horizon for MCR and RHCR covers  $k$  arrivals plus the arrivals which take place during processing if the oven would be loaded at the moment of the  $k$ th arrival. DJAH includes the next arrival for all product types which are to be processed and all other arrivals up to the moment a next machine becomes available. Note that arrival events of new data are not explicitly included in the above models as decision moments.

Taking the above classification as a starting point the heuristics mentioned will now be treated in detail in the next subsections. We will conclude with a discussion on the possibilities for extension of the existing look-ahead strategies to the multiple machines case.

### 2.1. Minimum Batch Size rule

The MBS-rule addresses the single product machine case. Optimization is reduced to an evaluation of the current situation. MBS compares the queue length

with a fixed minimum batch size. If the number ( $q$ ) of parts in queue is greater than or equal to the minimum batch size ( $B$ ) the oven is loaded, otherwise the decision is postponed:

$$\begin{aligned} & \text{if } q \geq B \\ & \text{then load the oven} \\ & \text{else wait} \end{aligned} \quad (1)$$

Deb and Serfozo (1973) show how to relate a minimum batch size to minimization of the expected continuously discounted cost or the expected averaged cost over an infinite horizon in a model with Poisson arrivals. As Fowler *et al.* (1992) remark, their computations can also be used to minimize the expected average waiting time in the case that the cost of serving customers is set to zero and the cost of customers waiting is linear. Given the latter criterion for optimization, Glassey and Weng (1991) found that a choice of  $B = 1$  (in their terminology: the non-relaxation case) results in a performance which is almost as good as for any other choice of  $B$ . It should be noted that although the MBS-rule is intended for single server systems, it can also be applied for multi-server systems, which process only one type of product.

Situations in which multiple product types have to be handled by a single machine or even multiple machines are not covered by the MBS-rule. In fact, it is unclear how MBS has to be adapted. Should e.g. a minimum batch size be associated with every product type or is the oven to be loaded if the sum of the queue lengths exceeds a certain minimum? Nevertheless, it would be useful to have such a rule at our disposal, in order to be able to show the added value of information about future arrivals for decision making. Therefore we adopted the rule supplied by Weng and Leachman (1993), as an alternative to MBS. According to this rule, which will be named MBSX, every time an oven cycle is completed, a new cycle is started right away if there are products in queue. The type of product chosen is the one which shows the longest queue length. In the case of a tie, the product which requires the shortest processing time is loaded into the oven. If this still leads to a tie, one chooses randomly.

## 2.2. Dynamic Batching Heuristic

Glassey and Weng (1991) looked for a control strategy which would utilize the available information on future arrivals in order to improve system performance. As a criterion for optimization they took the minimization of average waiting time over the planning interval  $(t_0, t_0 + T)$ . In this definition  $t_0$  represents the decision moment. For the single machine single product case the objective becomes:

$$\text{Minimize } q(t - t_0) + \sum_{t_0 < t_i < t} (t - t_i) + \sum_{t < t_i \leq t_0 + T} (t_0 + T - t_i) \quad (2)$$

The set of decision moments  $D_L$  contains  $t_0$  and the first  $L$  arrival moments as far as they fall within the planning interval (i.e.  $t_i \leq t_0 + T$ ) and within the oven capacity (i.e.,  $q + i \leq C$ ). By subtracting the objective value for the decision moment  $t = t_0$  it is found that this strategy is equivalent to the following rule:

$$\begin{aligned} & \text{If } q \geq C \\ & \text{then load the oven} \\ & \text{else wait until lot } k \text{ has arrived before loading the oven} \end{aligned}$$

$$\text{here } k = \underset{0 \leq i \leq i_{\max}}{\operatorname{argmin}} \{q(t_i - t_0) - i(t_0 + T - t_i)\} \quad (3)$$

with

$$i_{\max} = \min\{i', C - q\},$$

$$i' = \underset{d=1 \dots L}{\operatorname{argmax}} \{t_d | t_d \leq t_0 + T\},$$

$$0 < L \leq C - q$$

If the number in queue ( $q$ ) is greater than or equal to the oven capacity ( $C$ ) the decision is to load the oven right away. On the other hand, if the number in queue is smaller than the oven capacity, the heuristic decides upon the number of arrivals ( $k$ ) to wait for before loading the oven. Hence, instead of postponing a decision if more favourable conditions are foreseen, it forces jumping ahead. In this way some intermediate decision events are skipped.

In Fig. 2 the evaluation is graphically depicted for the situation where loading is postponed to the second arrival after  $t_0$ . The incurred waiting times are referred to as  $\text{Area}_1$ , while  $\text{Area}_2$  represents the saved waiting time. In case of a tie the earliest possible loading time is chosen. In fact the heuristic boils down to a comparison of the accumulated waiting time for a period  $T$  if the oven were to be loaded at  $t_0$  with the total waiting time over the same period if the oven were to be loaded at the moment of a future arrival.

Glassey and Weng (1991) have tested the performance of their heuristic in comparison with MBS under various conditions. In particular they compared DBH and MBS for different arrival patterns (Poisson and uniform distribution) and they investigated the robustness of the heuristic by replacing known data about future arrivals by forecasting data. The effects of using other settings for system parameters like oven capacity or processing time on system performance are not so clear. Moreover, they did not consider the case of missing data as a test for robustness. Finally it should be mentioned that Glassey *et al.* (1993) describe an extension of the DBH-heuristic to the multiple products multiple machines case, but an explicit formulation of the heuristics is not given.

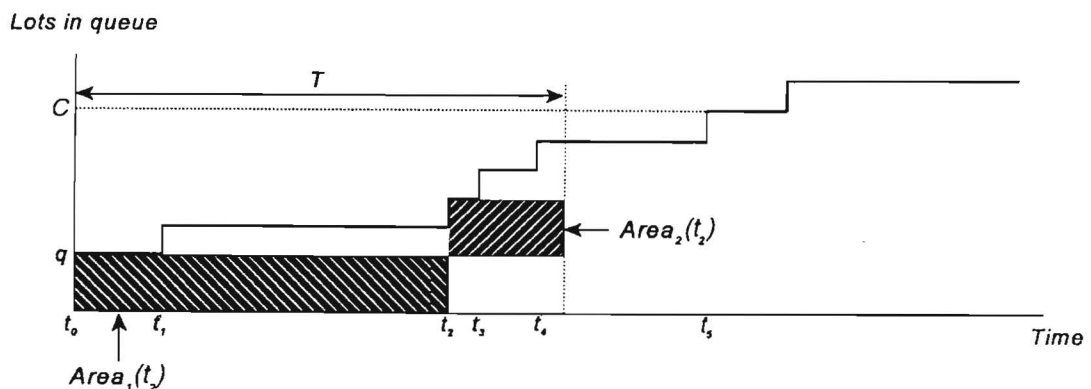


Figure 2. By loading the oven at  $t = t_2$  instead of  $t = t_0$  we create a waiting time of magnitude  $\text{Area}_1(t_2) = q(t_2 - t_0)$ . However, we annihilate a waiting time of magnitude  $\text{Area}_2(t_2) = 2(t_0 + T - t_2)$ .

### 2.3. Next Arrival Control Heuristic

Fowler *et al.* (1992) considered a heuristic where only the first future arrival (next arrival) is taken into account. They named their heuristic the Next Arrival Control Heuristic (NACH). For the single product single machine case their strategy is given by:

If  $q \geq C$   
 then load the oven  
 else if  $q(t_1 - t_0) - (t_0 + T - t_1) < 0$   
     then wait  
     else load the oven

(4)

For the single product single machine case NACH shows great resemblance to DBH for  $L = 1$ . What distinguishes NACH from DBH, though is its rolling horizon. If it seems more profitable to load the oven at the next arrival the decision is postponed. Note that now  $T$  plays the role of a planning parameter which is decoupled from the information horizon.

The heuristic is compared with DBH and MBS (Fowler *et al.* 1992) and shows especially good performance in the case where forecasting errors are introduced. The improvement of performance is to be ascribed to its rolling horizon, which makes it possible to include new or better quality information at a later moment. In this respect, NACH differs from DBH with  $L > 1$ .

Fowler *et al.* (1992) were the first to examine the multiple products single machine case. They adapted their heuristic to consider the next arrival for each of the product types. In the sequel we shall use the following notation for the multiple products single machine case:

N	The number of product types
J	The set of product type identifiers j
$C_j$	The oven capacity for products of type j
$T_j$	Processing time for products of type j
$q_j$	The number of products in queue at $t_0$ for products of type j
$t_{k,j}$	the $k$ th arrival of a product after $t_0$ for products of type j

In addition, we introduce for the NACH heuristic:

$D_j$	Total delay in case product j is loaded
SN	The set of product types for which it is not worthwhile to wait for a next arrival
SW	The set of product types for which it is worthwhile to wait for a next arrival
$W_j$	The total delay experienced by the other products during processing of the products of type j ( $T_j \sum_{i \neq j} q_i$ )
$NACH_1(j)$	Product j evaluated according to the NACH heuristic for the single product case

The NACH heuristic which addresses the multiple products single machine case is given in (5) below:

*Situation 1.*

The oven becomes idle:  
 If full loads are available

then select a product  $j^* = \underset{q_j \geq C_j}{\operatorname{argmin}} W_j$  and load the oven  
 else evaluate  $NACH_1(j)$  for all  $j$ ,  
     if  $SW = J$   
         then wait  
     else if  $SN = J$  then select  $j^* = \underset{j \in J}{\operatorname{argmin}} W_j$   
         else  
 define  $D_j = \begin{cases} W_j + \sum_{i=1}^N \max(0, t_0 + T_j - t_{1,j}) & j \in SN \\ \sum_{i=1}^N (q_i(t_{1,j} - t_0)) + W_j + \sum_{i=1}^N \max(0, t_{1,j} + T_j - t_{1,i}) & j \in SW \end{cases}$   
     select a product with  $j^* = \underset{j \in J}{\operatorname{argmin}} D_j$   
     if  $j^* \in SN$   
         then load the oven  
     else wait

(5)

## Situation 2

The oven is idle and a product type  $j$  arrives, then proceed as indicated by  $NACH_1(j)$

What is striking in the formulation is the separate treatment of the two possible decision situations, namely new oven availability events and new arrival events for an idle oven. This is in contrast with the single product case where a single logic is used.

In the first situation a distinction is made between a setting in which one or more full loads are available for one of the product types  $j$  in the set of product types  $J$ , and a setting where there are no full loads. Its purpose is to prevent very long waiting times for product types which have relatively long processing times. This was recognized by Conway *et al.* (1967) as a major disadvantage of a Shortest Processing Time rule (SPT). The set of options is restricted to those products for which  $q_j \geq C_j$ .

If full loads are available the oven is loaded right away. The product type to be loaded is determined using a Weighted Shortest Processing Time scheduling rule (WSPT). According to this rule the product type  $j^*$  is chosen which is associated with minimal delay  $W_j$ , given the information on queue lengths  $q_j$  and the respective processing times  $T_j$ , only.

If no full loads are available, first each product type  $j$  is evaluated using the NACH heuristic for the single product case  $NACH_1(j)$ , ignoring the other products. The decision options for a product are restricted according to the outcome of  $NACH_1$ . If  $NACH_1(j)$  gives waiting as its outcome, then product  $j$  is put in class SW for which waiting is the only option, else in class SN for which start processing is the only option. In the case where  $NACH_1$  indicates for all products to wait ( $SW = J$ ), the decision is postponed to a next arrival, while if according to  $NACH_1$  all products should start now ( $SN = J$ ), WSPT is again applied. This leaves the case in which there are some product types in SW and on the other hand a number of product types in SN. In order to determine the most favourable option the total delay  $D_j$  for each of the options is computed. In the case where  $j$  belongs to SN, the waiting time is computed as the summation of  $W_j$  and the waiting times for each of the next arrivals  $t_{1,j}$  provided they fall within the processing time. In the other case computations include the waiting times which are caused by delaying the loading of the oven until the next arrival of a product type  $j$ . If minimal delay



is found for a product type which should be loaded now, a new oven cycle starts, otherwise the decision is delayed until a next arrival.

The logic which is called upon if the decision moment coincides with an arrival is much simpler. It considers only the type of product which arrived, without accounting for the interaction with other product types. In order to decide if the oven should be loaded the  $NACH_1$  heuristic is called upon.

What is remarkable in the formulation of the NACH heuristic for the multiple products case is the fact that no single criterion is applied for optimization. Although it can be concluded that optimization is directed towards the minimization of the expected waiting time, different look-ahead horizons are applied. While in the case of full loads only the current situation is taken into account, in settings where  $D_j$  has to be computed also the first arrivals for all product types are included in the computations. Note that intermediary arrivals are left out (e.g. a second arrival of a certain product type which arrives before the first arrival of a product of another type). Another example is found in situation 2. Only the first arrival of the product type in question is considered, leaving out information on the current situation and future arrivals with regard to other product types.

A second remark concerns the seemingly unnecessary restrictions which are imposed upon the options open to the decision maker. While in situation 1 the choice to assign priority to full loads is well motivated, the subdivision of product types in two sets (SN, SW) in the case where no full loads are available is questionable. Why should options be reduced based upon the application of  $NACH_1$ , which does not take information on other product types into account? It seems more logical to determine minimal delay by computing  $(D_j)$  for every product in settings where the oven is loaded right away as well as settings where the loading of the oven is postponed. A second example is found in situation 2 where options are limited to just one product type. Why not evaluate all product types when new information on future arrivals is available?

#### 2.4. Minimum Cost Rate heuristic; Rolling Horizon Cost Rate heuristic

In response to the NACH heuristic Weng and Leachman (1993) devised a Minimum Cost Rate heuristic. They derive a scheduling rule (6) for the single product single machine case which is similar to the discrete time dynamic demand lot sizing heuristic proposed by Silver and Meal (1973).

*If  $q \geq C$   
then load the oven  
else wait until lot  $k$  has arrived before loading the oven*

$$\text{with } k = \underset{0 \leq i \leq i_{\max}}{\operatorname{argmin}} \frac{TC(t_i)}{t_i + T - t_0}$$

$$\text{where } i_{\max} = \max \{0, C - q\}$$

$$TC(t_i) = q(t_i - t_0) + \sum_{t_0 < t_d < t_i} (t_i - t_d) + \sum_{t_i < t_d \leq t_i + T} (t_i + T - t_d) \quad (6)$$

Unlike DBH, the decision options are not limited by a fixed horizon or a look-ahead number but only by the capacity of the oven. For each of the possible starting moments  $t_i$  (including  $t_0$ ) the average queue length is evaluated over the period starting at  $t_0$  and ending at  $t_i + T$ , when the oven cycle is completed. The starting



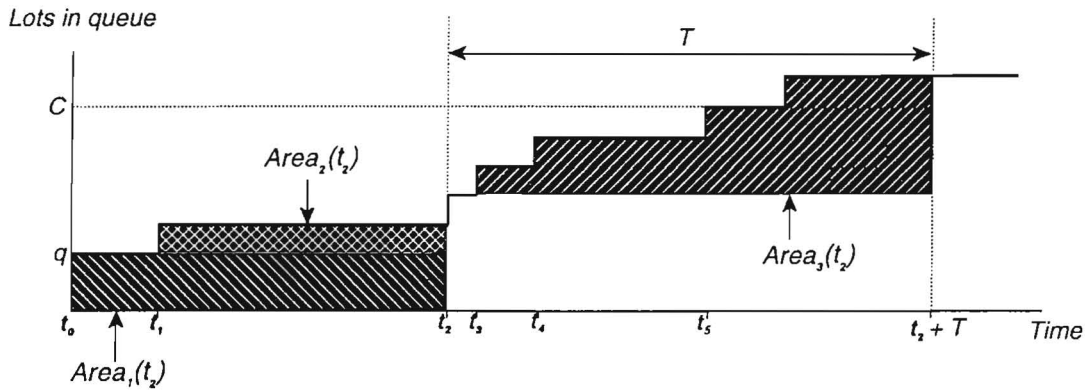


Figure 3. In the case where the oven is loaded at  $t_2$  the accumulated waiting time (TC) consists of the waiting time for the items in queue at the decision moment ( $\text{Area}_1(t_2) = q(t_2 - t_0)$ ), the items arriving before  $t_2$  ( $\text{Area}_2(t_2) = t_2 - t_1$ ) and the items, which arrive during processing of the batch ( $\text{Area}_3(t_2) = \sum (t_2 + T - t_k)$ ). The average queue length is found by dividing TC by the period  $t_2 + T - t_0$ .

moment which coincides with the minimum average queue length is chosen. Note that MCR forces jumping, just like DBH. Via Little's Law (1961) there is a relation with the minimization of the average delay. In case of a tie the earliest possible loading time is chosen. The evaluation is shown in Fig. 3 for  $t_2$ .

Although the MCR heuristic is intended to minimize average delay, it can easily be adapted for minimization of a more general cost criterion. To represent the cost structure found in the practical business situation the total cost function is reformulated as:

$$TV(t) = \Phi + TC(t) \quad (7)$$

In the formula a linear relationship is supposed to exist between waiting time and its associated costs, e.g. inventory costs. In addition to costs associated with waiting a cost function  $\Phi$  is introduced to cover the types of operating costs. It is assumed that these costs constitute a fixed amount due to internal transportation of parts, the allocated manpower and the consumed energy.

Extension of the MCR-heuristic to the multiple products case is straightforward. Evaluation for a certain product type only involves the inclusion of the accumulated waiting times for the other types of products, while the same basic logic derived from the Silver and Meal heuristic can be applied. Another addition is a rule also found in the NACH heuristic which gives preference to full loads. Writing  $TC_{ij}(t)$  for the total waiting costs for product  $i$  if products of type  $j$  are loaded at  $t$ , this rule can be stated as follows:

*If full loads are available  
then select a product*

$$j^* = \underset{q_j \geq C_j}{\operatorname{argmin}} \frac{\sum_{i=1}^N TC_{ij}(t_0)}{T_j}$$

*and load the oven  
else wait until  $k^*$  products of type  $j^*$  have arrived before loading the oven  
with*

$$(j^*, k^*) = \underset{\substack{(j,k): j \in J, \\ 0 \leq k \leq C_j - q_j}}{\operatorname{argmin}} \frac{\sum_{i=1}^N TC_{ij}(t_{k,j})}{t_{k,j} + T_j - t_0} \quad (8)$$

$$TC_{ij}(t) = \begin{cases} T_j \max(q_i - C_i, 0) + \sum_{t_0 < t_{d,i} < t + T_j} (t_0 + T_j - t_{d,i}) & t = t_0, i = j \\ q_i(t - t_0) + \sum_{t_0 < t_{d,i} < t} (t - t_{d,i}) + \sum_{t < t_{d,i} < t + T_j} (t + T_j - t_{d,i}) & t > t_0, i = j \\ q_i(t + T_j - t_0) + \sum_{t_0 < t_{d,i} < t + T_j} (t + T_j - t_{d,i}) & t \geq t_0, i \neq j \end{cases}$$

Note that in (8) a distinction is made between the cost functions (TC) in case the oven is loaded at  $t_0$  with a product of type  $j$ , and a situation in which the oven is loaded at a next arrival ( $t_{i,j}$ ). The waiting costs for the other products are computed according to the last cost function. The product which is associated with a minimum value for average queue length ( $j^*$ ) is chosen to be loaded next. Loading may be instantly ( $k^* = 0$ ), but may also require waiting for a future arrival ( $k^* > 0$ ).

If cost minimization is taken as a criterion the total cost function should be reformulated for the multiple products case as:

$$TV_{ij}(t) = \Phi_j + TC_{ij}(t) \quad (9)$$

In (9) it is reflected that operating costs for each of the product types may be different due to their physical characteristics which require specific handling during internal transport or processing. For reasons of simplicity and clarity of exposition, waiting costs are assumed to be uniform for all types of products.

The MCR-heuristic was tested (Weng and Leachman 1993) in comparison with MBS, DBH and NACH. It performs slightly better than DBH and NACH for moderate traffic intensity in the single product case. For situations where multiple product types have to be processed, MCR shows better performance than NACH. This is what one would expect, since the MCR heuristic uses more information. The simulations performed by Robinson *et al.* (1995) show, though, that MCR is not a very robust heuristic. Therefore, they adapt MCR by incorporating a rolling horizon into the cost rate function: if the cost rate function indicates that it is worthwhile to wait, it is decided to postpone the decision instead of jumping ahead to a future arrival. The performance of the Rolling Horizon Cost Rate heuristic (RHCR) was compared with MBS, NACH and MCR (Robinson *et al.* 1995). For both the single product and the multiple products case, results for MCR and RHCR are almost identical if there is no error in the prediction of future arrivals. If forecasting errors are introduced, RHCR outperforms MCR, but it shows no better results than NACH.

## 2.5. A new approach: the Dynamic Job Assignment Heuristic

We now propose a new look-ahead strategy which we named *Dynamic Job Assignment Heuristic* (DJAH). Our strategy tries to combine the strong points of the NACH and MCR heuristics in order to improve the systems performance. While NACH proves to be a very robust heuristic which only needs little data on future arrivals, MCR has the advantage of a clearly defined criterion for optimization. For the single product single machine case we formulate DJAH as:

if  $q \geq C$   
 then load the oven  
 else

$$\text{if } \frac{1}{q} TV(t_0) > \frac{1}{q+1} TV(t_1)$$

then wait

else load the oven

with

$$\begin{aligned} TV(t_0) &= \Phi + \sum_{t_0 < t_k < H^0} (H^0 - t_k) \\ TV(t_1) &= \Phi + q(t_1 - t_0) + \sum_{t_1 < t_k < H^1} (H^1 - t_k) \\ H^0 &= t_0 + T \\ H^1 &= t_1 + T \end{aligned} \quad (10)$$

The reduced criterion for optimization for DJAH is analogous to the MCR heuristic, except for the fact that costs (TV) are not minimized per unit of time but per item of the batch. This seems a more natural choice, because production costs per item are more relevant than production costs per unit of time from a business point of view. The ‘proof of the pudding is in the eating’ as shown in §4: especially for settings where multiple products are handled and processing times are differentiated per product, simulation results indicate significantly better performance for DJAH in comparison with MCR.

Just like NACH, DJAH adopts a rolling horizon approach. It only includes the next arrival for each product type in its evaluation as alternative options to loading the oven right away. As opposed to NACH though, the waiting costs are not only determined by these next arrivals, but they also include other arrivals up to the time the next oven completes its cycle (i.e. the *cost horizon*  $H^0, H^1$ ). Hence, the options are decoupled from the information horizon in this DJAH approach. Figures 4a, b show the waiting costs which are taken into account for a setting where the oven is loaded at  $t_0$  and a setting where it is considered to load the oven at the next arrival ( $t_1$ ). Area<sub>1</sub> marks the incurred delay for items in queue. The accumulated waiting time up to the moment the machine completes its cycle is depicted as Area<sub>2</sub>.

It should be mentioned that although the specific cost criterion was chosen here, the basic logic underlying DJAJ leaves the possibility to introduce other cost functions to represent specific cost structures found in business, such as penalty costs for late deliveries or maintenance costs. According to Blackstone *et al.* (1982) and our own experience such flexibility is really worthwhile, since most rules are dominated by non-cost performance measures like flow time, lateness and tardiness.

A simple extension suffices to cover the multiple products case:

*If full loads are available*  
 then select product  $j^* = \underset{q_j \geq C_j}{\operatorname{argmin}} \frac{TV_j(t_0)}{C_j}$  and load the oven  
 else  
 if

$$\min_{\substack{j=1 \dots N \\ q_j > 0}} \frac{1}{q_j} TV_j(t_0) > \min_{j=1 \dots N} \frac{1}{q_j + 1} TV_j(t_{1,j})$$

then wait

else load the oven, select product

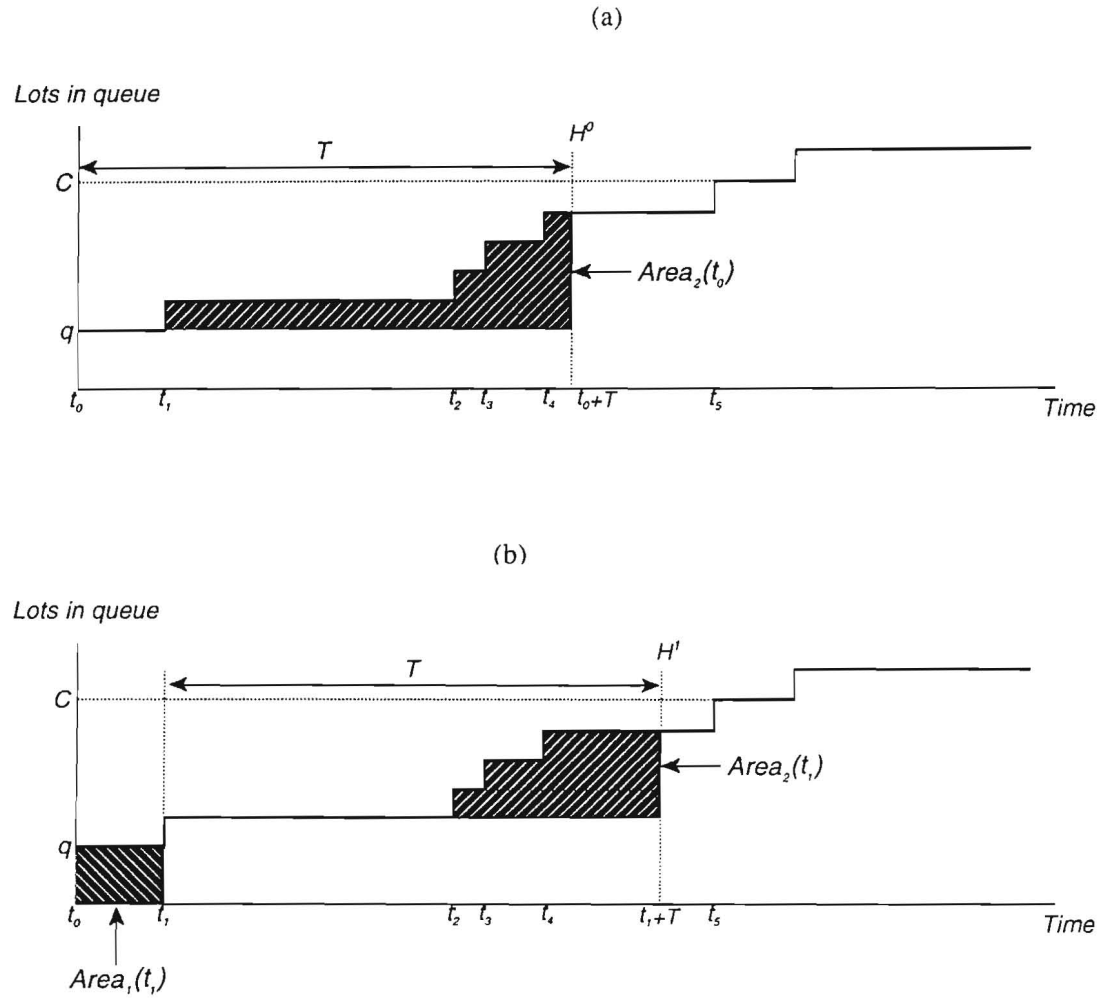


Figure 4a, b. Starting the oven at  $t_0$  causes waiting costs for the items, that arrive during processing ( $\text{Area}_2(t_0) = \sum (H^0 - t_k)$ ). On the other hand, if the oven were to be loaded at  $t_1$ , next to these costs ( $\text{Area}_2(t_1) = \sum (H^1 - t_k)$ ) also the waiting costs for items in queue ( $q$ ) at the decision moment ( $t_0$ ) should be taken into account ( $\text{Area}_1(t_1) = q(t_1 - t_0)$ ).

$$j^* = \underset{j=1 \dots N}{\operatorname{argmin}} \frac{1}{q_j} TV_j(t_0)$$

with

$$TV_j(t_0) = \Phi_j + T_j \max(q_j - C_j, 0) + T_j \sum_{i \neq j}^N q_i + \sum_{i=1}^N \sum_{t_0 < t_{k,i} < H_j^0} (H_j^0 - t_{k,i})$$

$$TV_j(t_{1,j}) = \Phi_j + q_j(t_{1,j} - t_0) + (H_j^1 - t_0) \sum_{i \neq j}^N (q_i) + \sum_{i=1}^N \sum_{t_0 < t_{k,i} < H_j^1} (H_j^1 - t_{k,i}) - T_j$$

$$H_j^0 = t_0 + T_j$$

$$H_j^1 = t_{1,j} + T_j$$

(11)

Again, just like MCR and NACH, priority is assigned to product types for which a full load is available. The product type which shows the lowest costs per item (cost price) over the cost horizon ( $H^0$ ) is chosen among those products for which there is a full load. In all other cases it is decided to load the oven if no lower cost price is expected to be realized at the time of the first future arrival for one of the distinguished product types. Note that costs in case the oven would be loaded at a next arrival  $t_{1,j}$  ( $TV_j(t_{1,j})$ ) are adjusted for the fact that no waiting costs are incurred for the product of type  $j$  by subtracting  $T_j$ .

An extension of DJAH is also applicable for system configurations which consist of multiple machines. Here we limit ourselves to its application for identical machines. Configurations of multiple non-identical batch processing machines are a subject of our ongoing research (van der Zee 1997). Given the assumption that only identical machines are available, there is no need to augment decision options to include alternative machines. After all, the use of another machine does not result in lower costs. As a consequence, the extension of DJAH to the multiple machines case is straightforward. For systems which consist of a number of identical machines that handle only one type of product (10) has to be reformulated as:

If  $q \geq C$   
 then load the oven  
 else  
 if  $\frac{1}{q} TV(t_0) > \frac{1}{q+1} TV(t_1)$   
 then wait  
 else load the oven  
 with

$$TV(t_0) = \Phi + \sum_{t_0 < t_k < H^0} (H^0 - t_k)$$

$$TV(t_1) = \Phi + q(t_1 - t_0) + \sum_{t_1 < t_k < H^1} (H^1 - t_k)$$

$$a_{\min} = \min_{t'_a = t_0} a$$

$$H^0 = H^1 = \min_{a \neq a_{\min}} t'_a \quad (12)$$

Note that the look-ahead horizon ( $H^0, H^1$ ) is adapted to account for the presence of multiple machines. Here  $a$  refers to the machine serial number. Further,  $t'_a$  denotes the moment machine  $a$  completes a cycle. Note that  $H^0 = H^1 = t_0$  if two or more machines are available at time  $t_0$ . The moment the next machine becomes available is determined by selecting the minimum of the cycle completion times  $t'_a$ . It is interesting to observe, that even if two or more machines are available at  $t_0$  and  $q > 0$ , it can happen that one decides to wait until  $t_1$ . This effect is due to the setup costs  $\Phi$ . If more than two machines are available at  $t_0$  the machine with the lowest serial number ( $a_{\min}$ ) is chosen as the one to be loaded first. It will be clear that this priority rule can simply be modified, e.g. to balance use of each of the machines, without influencing averaged cost price.

In a similar way, using (11), a control strategy is found for the multiple products multiple identical machines case (13). In (13) the cost horizon ( $H_j^0, H_j^1$ ) has been

adapted to account for cases in which the machine processing the product of type  $j$  becomes available earlier than any other machine, as a consequence of a short processing time. Just like the single machine case, the cost function  $TV_j(t_{1,j})$  is adjusted for the fact that no waiting costs are incurred for the first arrival of a product of type  $j$ . Notice that the adjustment  $\max(H_j^1 - t_{1,j}, 0)$  also accounts for situations in which the next machine is available before the oven cycle ends.

It should be mentioned that the extension to multiple machines could also be easily made to NACH. However, we do not pursue this option in view of NACH's relatively poor performance (see §4) and the fact that NACH is less suitable for including setup costs. For MCR/RHCR it is less clear how extensions to the multiple machines case have to be made. For example, should all machines be included in the schedule, or is scheduling limited to those machines available at the decision moment? On the other hand, one should ask oneself if extension is worthwhile if the performance of MCR/RHCR is taken into account, given its information intensity. We will come back to this point in §4.

*If full loads are available  
then select product*

$$j^* = \operatorname{argmin}_{q_j \geq C_j} \frac{TV_j(t_0)}{C_j}$$

*and load the oven  
else  
if*

$$\min_{\substack{j=1 \dots N \\ q_j > 0}} \frac{1}{q_j} TV_j(t_0) > \min_{j=1 \dots N} \frac{1}{q_j + 1} TV_j(t_{1,j})$$

*then wait  
else load the oven, select product  $j^* = \operatorname{argmin}_{j=1 \dots N} \frac{1}{q_j} TV_j(t_0)$   
with*

$$\begin{aligned} TV_j(t_0) &= \Phi_j + (H_j^0 - t_0) \max(q_j - C_j, 0) + (H_j^0 - t_0) \sum_{i \neq j}^N q_i + \sum_{i=1}^N \sum_{t_0 < t_{k,i} < H_j^0} (H_j^0 - t_{k,i}) \\ TV_j(t_{1,j}) &= \Phi_j + q_j(t_{1,j} - t_0) + (H_j^1 - t_0) \sum_{i \neq j}^N q_i \\ &\quad + \sum_{i=1}^N \sum_{t_0 < t_{k,i} < H_j^1} (H_j^1 - t_{k,i}) - \max(H_j^1 - t_{1,j}, 0) \end{aligned}$$

$$a_{\min} = \min_{t'_a = t_0} a$$

$$H_j^0 = \min \left( \min_{a \neq a_{\min}} t'_a, t_0 + T_j \right)$$

$$H_j^1 = \min \left( \min_{a \neq a_{\min}} t'_a, t_{1,j} + T_j \right) \quad (13)$$

### 3. Comparison of the heuristics: design of a simulation study

To show the potential of the newly developed DJAH heuristic it was extensively tested by a series of simulation experiments. Its response for various system configurations, which reflect different settings for product specifications and number of machines, was analysed in comparison with the other known heuristics.

The simulations which were carried out concern single/multi-product and single/multi-machine configurations. To enable judgement on the relative performance of the heuristics, for each of these cases different settings were studied. To allow for comparison of simulations, our settings have been derived from cases already mentioned in the literature. In Table 2 an overview of the simulation experiments is given.

Two different criteria were applied to analyse the performance of the control strategies: the average flow time criterion and the minimal cost criterion. While the first criterion was applied to all control strategies, the latter criterion was only applied to MBS, MCR and DJAH, because NACH and DBH do not consider logistical costs nor can they easily be extended to do so (see §2). For all simulations operating costs were chosen uniformly: waiting costs equal 1 per unit of time, while

Factor	Configuration			
	N = 1, M = 1	N = n, M = 1	N = 1, M = m	N = n, M = m
(1) Criterion	Minimal average flow time; Minimal cost, ( $\Phi = 60$ )	Minimal average flow time; Minimal cost, ( $\Phi = 60$ )	Minimal average flow time; Minimal cost, ( $\Phi = 60$ )	Minimal average flow time; Minimal cost, ( $\Phi = 60$ )
(2) Control strategy	MBS DBH NACH MCR DJAH	MBSX NACH MCR DJAH	MBS DJAH	MBSX DJAH
(3) Interarrival distribution	<b>Exponential</b> Uniform	<b>Exponential</b> Uniform	<b>Exponential</b> Uniform	<b>Exponential</b> Uniform
(4) Quality of Information (with regard to future arrivals)	<b>Known</b> Predicted, Missing data	<b>Known</b> Predicted, Missing data	<b>Known</b> Predicted, Missing data	<b>Known</b> Predicted, Missing data
(5) Number of product types (N)		2		4
(6) Product mix in percentages( $s_j$ )		(50:50), (75:25)		(25:25:25:25), (50:30:10:10)
(7) Capacity per product ( $C_j$ )	5, 10	(5,5), (7,3)	5, 10	(5,5,5,5), (8,6,4,2)
(8) Processing time per product ( $T_j$ )	25, 50	(25,25), (40,10)	25, 50	25,25,25,25), (40,30,20,10)
(9) Number of machines (M)			2, 4	2
(10) Traffic intensity ( $\rho$ )	0.3 0.6 0.9	0.3 0.6 0.9	0.3 0.6 0.9	0.3 0.6 0.9
Number of simulations	132	126	57	57

Table 2. Design of the simulation study.



setup costs constitute a fixed amount of 60. Note that we did not include RHCR in the simulations. The reason for this is that research by Robinson *et al.* (1995) already showed that if there is no error in the prediction of future arrivals, simulation results for MCR and RHCR are almost identical. On the other hand they showed that if forecasting errors are introduced, RHCR yields no better results than NACH (see also subsection 2.4).

For each scenario one basic setting has been defined which reflects a particular setting for product and machine characteristics. These settings are marked boldly in Table 2. Alternative system configurations were chosen by changing the value for exactly one of these decision variables. For example, to estimate the effect of a longer processing time on the system performance in the single product machine case, the processing time was set to 50 time units instead of 25. In the same way the robustness of the heuristics, reflected by its response to forecasting errors or incomplete data on future arrivals, was evaluated. Forecasting errors are assumed to be normally distributed with mean equal to zero and a standard deviation, which equals half the standard deviation of the interarrival times ( $\sigma$ ). In the simulation model forecasting errors are associated with the data the decision maker receives on future arrival moments. Note that the possibility of forecasting errors requires a more refined updating of the information set on future arrivals (AR). At each decision moment corresponding with a product arrival the forecast arrival moment for this product is removed from AR. Further, for decision making arrivals that are forecast at  $t$ , but in 'reality' occur at  $\tilde{t}$  later than  $t$ , are ignored once the decision moment  $t_0$  passes  $t$ .

Also performance for heuristics in situations in which the decision maker lacks on average 50% of the data on future arrivals is tested. These situations are modelled by associating a chance of 0.5 with each arriving product that it is not reported to the decision maker before its actual arrival. As a consequence the heuristics have to base their decision on the knowledge of later arrivals. For example, in case of the NACH heuristic the lack of information on the next arrival may mean that a decision is based on its knowledge of a second or even third arrival. This kind of robustness test is very important because many practical situations in business are characterized by the fact that only incomplete or imprecise information is available to support decision making.

Because workload tends to have a major impact on the performance of a queueing system, all settings mentioned were analysed for low (30%), moderate (60%) and high (90%) traffic intensities. The traffic intensity ( $\rho$ ) is defined (Chaudry and Templeton (1983)) as:

$$\rho = \frac{\text{mean arrival rate of customers}}{\text{maximum service rate when operating at maximum capacity}} = \lambda \sum_{j=1}^N \frac{s_j T_j}{M C_j} \quad (14)$$

The maximum service rate is determined by the product mix ( $s_j$ ), the processing times for each product type  $j$  ( $T_j$ ), the machine capacities per product type ( $C_j$ ) and the number of machines ( $M$ ). Given a specific setting for traffic intensity, (14) can be used to determine the mean arrival rate  $\lambda$ . This variable is used to specify the interarrival distribution. While the Poisson distribution is fully defined by  $\lambda$ , the uniform distribution is characterized by its range. To establish a mean interarrival time  $1/\lambda$  we chose the range  $[0.5/\lambda, 1.5/\lambda]$ .

The package which was used to carry out the simulation experiments is ExSpect (Bakkenist 1994). ExSpect is a Petri nets-based analysis tool. It allows for structural analysis as well as dynamic analysis by simulation. To facilitate the modelling process a logistic reference model was adopted (van der Zee 1995). A simulation model built according to this reference model can easily be adapted to incorporate new control rules or even new control structures. Moreover, the principles of object oriented design (see e.g. Booch 1994) underlying both ExSpect and the reference model, guarantee reusability of model components in order to support further research.

The performance of each heuristic was estimated using the batch means method (see e.g. Hoover and Perry 1989, Law and Kelton 1991). The simulation horizon is 775 000 time units, which allows for 30 batches of 25 000 time units, as the first batch is discarded to account for any start up bias. Each batch corresponds to several thousands of products completed in order to guarantee that the batch means will be approximately uncorrelated. A useful tool to examine the correlation of the batches is the runs test (see e.g. Hoover and Perry 1989, Gibbons and Chakraborti 1992). In our case, even for high traffic intensities, this test showed no significant correlation, given a significance level  $\alpha = 0.05$ . The standard deviation of the average waiting time emerging from the 30 batches is low for traffic intensities of 30 and 60% (typically in the order of 0–0.3% of the average waiting time). As may be expected standard deviation is higher for traffic intensities of 90% (in the order of 1.5% of the average waiting time). Our choice of simulation conditions is motivated by the experimental settings of Fowler *et al.* (1992).

#### 4. Analysis of simulation results

In the previous section a series of simulation experiments were designed. Now the results of these simulations will be discussed. First, attention will be paid to the performance in the case where average flow time is taken as a criterion for optimization. Next, simulation results for a minimal cost criterion will be treated. It should be noticed that settings for the simulation experiments will be indicated by the non-default factor which is studied, as mentioned in § 3 (Table 2).

##### 4.1. Average flow time

###### 4.1.1. Single product, single machine

In table 3 an overview is given of simulation results for a number of settings. The second and the third columns indicate the specific setting by mentioning the non-default factor which is studied and traffic intensities. The next five columns show the performance for each heuristic given a certain setting. To get an impression of the potential of the DJAH heuristic the last four columns present the percentage differences between DJAH and the other heuristics. Waiting times are presented with an accuracy of two decimals. The accuracy is determined by the design of the simulation study (see § 3). A positive difference indicates that DJAH performs better. To test the statistical validity of the differences a paired-*t* approach was used with a 95% confidence interval (see e.g. Law and Kelton 1991). Differences which are not statistically significant (typically between –0.5% and 0.5%) are not mentioned in the table.

The results are consistent with former simulation experiments found in the literature: the profit to be gained by adopting a heuristic which uses knowledge on future arrivals, instead of MBS, is large. The relative advantage of look-ahead

No.	Factor	$\rho$	MBS	NACH	DBH	MCR	DJAH	$\Delta_1$	$\Delta_2$	$\Delta_3$	$\Delta_4$
1	Default	0.3	10.96	6.25	6.32	6.06	6.12	44.7	2.1	3.2	-1.0
2	Default	0.6	13.97	9.80	9.83	9.71	9.72	30.4	0.8	1.1	-
3	Default	0.9	30.88	28.39	28.42	28.66	28.57	7.5	-0.6	-0.5	0.3
4	Uniform arrivals	0.3	8.77	7.49	7.49	7.26	7.48	14.7	-	-	-3.0
5	Uniform arrivals	0.6	11.04	9.96	9.92	9.73	9.83	11.0	1.3	0.9	-1.0
6	Uniform arrivals	0.9	11.93	11.24	11.23	11.21	11.17	6.4	0.6	0.5	0.4
7	Capacity	0.3	12.07	8.50	8.51	8.22	8.34	30.9	1.9	2.0	-1.5
8	Capacity	0.6	12.79	10.49	10.42	10.23	10.47	18.1	-	-0.5	-2.3
9	Capacity	0.9	21.04	19.11	18.89	19.04	19.29	8.3	-0.9	-2.1	-1.3
10	Service time	0.3	21.91	12.50	12.63	12.09	12.20	44.3	2.4	3.4	-0.9
11	Service time	0.6	27.31	19.59	19.58	19.35	19.41	28.9	0.9	0.9	-
12	Service time	0.9	65.29	60.57	60.52	61.19	61.01	6.6	-0.7	-0.8	-
13	Data missing	0.3	10.96	8.23	8.61	8.62	8.17	25.5	0.7	5.1	5.2
14	Data missing	0.6	13.97	11.48	11.86	11.82	11.43	18.2	0.4	3.6	3.3
15	Data missing	0.9	30.88	29.92	30.24	30.25	29.97	2.9	-	0.9	0.9
16	Data forecast	0.3	10.96	9.70	9.66	9.72	9.70	11.5	-	-	-
17	Data forecast	0.6	13.97	11.53	11.55	11.61	11.58	17.1	-0.4	-	-
18	Data forecast	0.9	30.88	29.41	29.45	29.79	29.68	3.9	-0.9	-0.8	-

$$\Delta_1 = 100 \cdot (\text{MBS} - \text{DJAH}) / \text{MBS}$$

$$\Delta_2 = 100 \cdot (\text{NACH} - \text{DJAH}) / \text{NACH}$$

$$\Delta_3 = 100 \cdot (\text{DBH} - \text{DJAH}) / \text{DBH}$$

$$\Delta_4 = 100 \cdot (\text{MCR} - \text{DJAH}) / \text{MCR}$$

Table 3. Average waiting time for  $N = 1, M = 1$ .

strategies over MBS tends to be larger if arrival patterns are more irregular. This is illustrated in Table 3 where differences for uniform arrivals are smaller than for Poisson arrivals. It should be mentioned that for all settings the optimal MBS was chosen. Although Glassey and Weng (1991) suggest that in the case of Poisson arrivals a choice for a minimum batch size of 1 would (nearly) equal results for the best MBS, it is found that differences may be in the order of 3–5%. Choosing  $\text{MBS} = 1$  might therefore not always result in a good approximation of the best possible performance for MBS.

Another conclusion from former research was that differences between heuristics, except for MBS, are typically small. Although this is true for most settings presented here, the robustness test on missing data in particular shows that this might not always be the case. NACH and DJAH clearly have an advantage over DBH and MCR. This advantage is to be ascribed to their rolling horizon approach which leaves room to include new or updated information on future arrivals.

For higher oven capacity, differences between MBS and the other heuristics are smaller. This seems consistent with the fact that the attraction of waiting for a next arrival compared with that of loading the oven right away tends to get smaller as the average number of products in queue at the time of the decision will grow.

The influence of service time on average waiting time is as expected: doubling the service time, while halving the arrival rate ( $\lambda$ ), results in roughly twice the average waiting time. Although the number of products arriving during service is the same for both cases, products have to wait twice as long before the oven cycle is completed. It leaves the relative performance of the heuristics unharmed.

In general the results favour MCR for settings in which full knowledge is available about future arrivals up to the moment marked by its information horizon. MCR shows marginally better results for low traffic intensities or higher oven capacities. On the other hand, if no such knowledge is available, DJAH is the best candidate, given its robustness and the fact that it bases its decisions on less information than MCR. Especially for low traffic intensities this means that the use of MCR requires the ability to look ahead over a long period (namely until the maximum batch size is reached).

#### *4.1.2. Multiple products, single machine*

The introduction of multiple product types that cannot be batched complicates the problem. Not only does one have to determine when to load the oven, but also the type of product to be loaded has to be determined. As a consequence of the larger product assortment which has to be handled by a single machine, higher average waiting times are to be expected. Moreover, less profit is to be gained by postponing the loading of the oven because other products will have to wait. These ideas are confirmed by a number of simulation experiments. The results for these simulations are shown in Tables 4a, b. Table 4a presents the average waiting time for the product mix, hence weighting of averaged product waiting times took place (according to products' share in the product mix). Table 4b shows waiting times per product type.

The first six settings mentioned in Table 4a are comparable with the first six settings for the single product single machine scenario. Results for these settings show that average waiting time is strongly influenced by the number of product types. It almost doubles for low and moderate traffic intensities. The fact that differences between the results for Poisson arrivals and for a uniform arrival pattern on the one hand and difference between performance of MBS and the look-ahead strategies on the other hand tend to diminish is also in agreement with the above propositions. After all, if it is more often decided to load the oven right away, the timing of the next arrival is becoming less relevant and less profit is realized by including information of this type.

The choice for a different product mix, where the first product has a dominant share, leads to a reduction of the average waiting time. Since the situation is more like the single product case this seems a logical consequence.

For settings where capacities depend on the product type, a distinction has to be made between the results for low traffic intensity on the one hand and moderate and high traffic intensities on the other hand. While for low traffic intensity the benefits of the higher oven capacity for products of type 1 are decisive, for moderate and high traffic intensities the small oven capacity for products of type 2 becomes a serious bottleneck (see also Table 4b). Note that a direct comparison of settings 10–12 with the default settings is not possible, because the maximum service rate is different from the maximum service rate for the default settings.

In general, product dependence of service times has a marginal impact on the system performance. Only for high traffic intensity the effect is noticeable. The 'loss' in situations where the length of an oven cycle is large is compensated by short oven cycles.

The robustness tests indicate that heuristics are less vulnerable to missing data or forecasting errors than in the single product type scenario. Again this can be explained by the fact that in general less profit is to be gained by postponing the decision to load the oven.

No.	Factor	$\rho$	MBSX	NACH	MCR	DJAH	$\Delta_1$	$\Delta_2$	$\Delta_3$
1	Default	0.3	18.04	13.40	13.04	12.94	28.3	3.4	0.8
2	Default	0.6	23.62	20.26	19.55	19.54	17.3	3.6	—
3	Default	0.9	43.01	40.60	39.75	39.86	7.3	1.8	—
4	Uniform arrivals	0.3	19.33	13.21	13.42	13.49	30.2	-2.1	—
5	Uniform arrivals	0.6	22.73	18.99	19.27	19.15	15.8	-0.8	0.6
6	Uniform arrivals	0.9	27.41	24.99	24.87	24.82	9.5	0.7	—
7	Product mix	0.3	16.74	11.96	11.85	11.64	30.5	2.7	1.8
8	Product mix	0.6	22.72	19.07	18.63	18.56	18.3	2.7	—
9	Product mix	0.9	42.59	39.87	39.20	39.06	8.3	2.0	—
10	Capacities	0.3	16.68	12.07	11.76	11.71	29.9	3.0	—
11	Capacities	0.6	26.07	21.59	21.12	20.97	19.6	2.9	0.7
12	Capacities	0.9	57.16	48.71	46.46	46.10	19.3	5.4	0.8
13	Service times	0.3	20.23	12.95	13.37	12.39	38.8	4.3	7.4
14	Service times	0.6	27.18	20.11	20.23	19.45	28.4	3.3	3.8
15	Service times	0.9	55.49	43.35	43.24	42.47	23.5	2.0	1.8
16	Data missing	0.3	18.03	15.46	15.52	15.12	16.2	2.2	2.6
17	Data missing	0.6	23.62	21.92	21.66	21.30	9.8	2.8	1.7
18	Data missing	0.9	43.01	41.84	41.23	41.08	4.5	1.8	—
19	Data forecast	0.3	18.03	16.41	16.73	16.45	8.8	—	1.7
20	Data forecast	0.6	23.62	21.94	21.68	21.55	8.8	1.8	0.6
21	Data forecast	0.9	43.01	41.55	41.10	41.01	4.7	1.3	—

$$\Delta_1 = 100 * (\text{MBSX} - \text{DJAH}) / \text{MBSX}$$

$$\Delta_2 = 100 * (\text{NACH} - \text{DJAH}) / \text{NACH}$$

$$\Delta_3 = 100 * (\text{MCR} - \text{DJAH}) / \text{MCR}$$

Table 4a. Average waiting time for  $N = 2, M = 1$ .

No.	MBS		NACH		MCR		DJAH	
	1	2	1	2	1	2	1	2
1	17.86	18.21	13.31	13.48	12.93	13.15	12.78	13.10
2	23.44	23.79	20.07	20.44	19.36	19.73	19.39	19.69
3	42.93	43.09	40.36	40.84	38.79	40.71	37.41	42.30
4	19.27	19.38	13.10	13.31	13.29	13.55	13.49	13.48
5	22.72	22.74	18.96	19.02	19.30	19.24	19.13	19.17
6	27.40	27.42	24.89	25.08	24.80	24.93	24.78	24.85
7	14.04	24.85	10.06	17.67	8.83	20.89	9.14	19.14
8	17.95	37.02	15.49	29.82	13.80	33.12	14.21	31.62
9	32.46	72.96	30.50	67.99	30.77	64.50	30.39	65.06
10	16.41	16.95	11.80	12.33	11.52	12.00	11.44	11.97
11	23.85	28.29	21.88	21.29	20.75	21.49	20.90	21.04
12	48.24	66.08	47.39	50.03	39.05	53.86	38.95	53.25
13	19.05	21.40	14.94	10.95	13.28	13.46	15.44	9.33
14	23.90	30.45	23.98	16.24	21.04	19.41	22.35	16.55
15	50.74	60.23	54.26	32.44	53.92	32.56	52.97	31.97
16	17.86	18.21	15.34	15.57	15.43	15.60	14.93	15.30
17	23.44	23.79	21.64	22.19	21.56	21.76	21.23	21.37
18	42.93	43.09	41.62	42.05	42.25	40.21	42.09	40.07
19	17.86	18.21	16.22	16.59	16.52	16.94	16.33	16.57
20	23.44	23.79	21.78	22.10	21.64	21.71	21.50	21.59
21	42.93	43.09	41.22	41.88	40.92	41.28	38.83	43.18

Table 4b. Average waiting time per product type for  $N = 2, M = 1$ .



The relative performance of DJAH is overall the strongest. Only for the setting where products arrive according to a uniform distribution NACH has an advantage over DJAH for low and moderate traffic intensities. It should be remarked that the results in Table 4a seem inconsistent with the outcomes of previous experiments by Weng and Leachman (1993). In their view MCR should perform better than NACH for product dependent service times. Therefore their settings were included in another series of simulation experiments for which the number of product types was raised to four. The results for these experiments are in agreement with the findings of Weng and Leachman. Apparently the performance of NACH depends on the number of product types.

#### 4.1.3. Single product, multiple machines

For the single product multiple machines case DJAH was compared to the best MBS strategy. Again it was found that a careful choice of the minimum batch size is necessary. Performance differences for  $MBS = 1$  and the optimal MBS strategy tend to become even greater, which further restricts the meaning of the proposition of Glassey and Weng (1991). Table 5 presents the results of the simulations.

Although results show great similarity with the results presented in Table 3, it is a remarkable fact that differences between MBS and DJAH for Poisson arrivals are significantly larger compared with the single product single machine case. For a larger number of machines, differences even go up, as can be concluded from the measurements for the last three settings.

No.	Factor	$\rho$	MBS	DJAH	$\Delta_1$
1	Default	0.3	5.86	2.83	51.7
2	Default	0.6	6.92	4.49	35.1
3	Default	0.9	14.21	13.29	6.5
4	Uniform arrivals	0.3	4.25	3.70	12.9
5	Uniform arrivals	0.6	5.34	5.05	5.4
6	Uniform arrivals	0.9	5.80	5.61	3.3
7	Capacity	0.3	6.23	3.95	36.6
8	Capacity	0.6	6.59	4.96	24.7
9	Capacity	0.9	9.93	9.05	8.9
10	Service time	0.3	11.75	5.66	51.8
11	Service time	0.6	13.81	9.02	34.7
12	Service time	0.9	28.73	26.94	6.2
13	Data missing	0.3	5.86	3.89	33.6
14	Data missing	0.6	6.92	5.43	21.5
15	Data missing	0.9	14.21	14.32	0.8
16	Data forecast	0.3	5.86	4.31	26.5
17	Data forecast	0.6	6.92	5.28	23.7
18	Data forecast	0.9	14.21	13.69	3.7
19	Machines av.	0.3	2.70	1.32	51.1
20	Machines av.	0.6	3.37	2.07	38.6
21	Machines av.	0.9	6.45	6.39	0.9

$$\Delta_1 = 100 * (MBS - DJAH) / MBS$$

Table 5. Average waiting time for  $N = 1, M = 2; 4$ .

No.	Factor	$\rho$	MBSX	DJAH	$\Delta_1$
1	Default	0.3	15.20	12.11	20.3
2	Default	0.6	20.95	18.16	13.3
3	Default	0.9	32.00	29.44	8.0
4	Uniform arrivals	0.3	15.69	13.12	16.4
5	Uniform arrivals	0.6	21.16	18.07	14.6
6	Uniform arrivals	0.9	26.54	23.25	12.4
7	Product mix	0.3	13.26	10.34	22.0
8	Product mix	0.6	19.41	16.74	13.8
9	Product mix	0.9	30.96	28.71	7.3
10	Capacities	0.3	10.35	8.11	21.6
11	Capacities	0.6	23.38	18.69	20.1
12	Capacities	0.9	47.97	34.58	27.9
13	Service times	0.3	14.62	11.58	20.8
14	Service times	0.6	20.88	17.82	14.7
15	Service times	0.9	33.99	30.30	10.9
16	Data missing	0.3	15.20	13.38	12.0
17	Data missing	0.6	20.95	19.19	8.4
18	Data missing	0.9	32.00	29.99	6.3
19	Data forecast	0.3	15.20	13.81	9.1
20	Data forecast	0.6	20.95	19.21	8.4
21	Data forecast	0.9	32.00	29.99	6.3

$$\Delta_1 = 100 * (\text{MBSX} - \text{DJAH}) / \text{MBSX}$$

Table 6. Average waiting time for  $N = 4$ ,  $M = 2$ .

#### 4.1.4. Multiple products, multiple machines

To study the multiple products, multiple machines scenario, simulations were performed for an example in which two machines process four types of products. The results for these simulation experiments conform to the results found for the other cases. Just as we saw earlier for the single machine case, it is found that an increase in the number of products which cannot be batched together corresponds with a strong increase of the average waiting time (compare Tables 5 and 6). This result suggests that in practical business situations it may be very worthwhile to reduce this number, e.g. by forming product families for which processing conditions are uniform. Of course this implicates that product specifications may have to be adapted.

### 4.2. Minimal cost

#### 4.2.1. Single product, single machine

To gain insight in the performance of look-ahead strategies for situations where minimization is aimed at operating costs we shall compare MBS, MCR and DJAH. As was explained in § 3, two types of operating costs are considered: setup costs and waiting costs. In Table 7a the results are presented of the simulations for the single product single machine scenario, given the new criterion.

The settings mentioned in Table 7a are identical to those for the average flow time criterion. To understand how the cost price is influenced by waiting costs on the one hand, and setup costs on the other hand, the corresponding values for average waiting time and average batch size are presented in Table 7b.



No.	Factor	$\rho$	MBS	MCR	DJAH	$\Delta_1$	$\Delta_2$
1	Default	0.3	37.26	30.64	30.25	18.8	1.3
2	Default	0.6	29.71	25.47	25.70	13.5	-0.9
3	Default	0.9	42.88	40.99	41.00	4.4	-
4	Uniform arrivals	0.3	36.69	35.51	35.37	3.6	0.4
5	Uniform arrivals	0.6	27.59	26.74	26.79	2.9	-0.2
6	Uniform arrivals	0.9	23.93	23.71	23.73	0.8	-
7	Capacity	0.3	28.30	22.96	23.57	16.7	-2.7
8	Capacity	0.6	22.18	18.86	19.31	12.9	-2.4
9	Capacity	0.9	27.17	25.30	25.57	5.9	-1.0
10	Service time	0.3	51.12	39.36	38.64	24.4	1.8
11	Service time	0.6	44.37	35.87	36.05	18.8	-0.5
12	Service time	0.9	77.29	73.54	73.25	5.2	0.4
13	Data missing	0.3	37.29	36.71	34.48	7.5	6.1
14	Data missing	0.6	29.71	28.77	27.96	5.9	2.8
15	Data missing	0.9	42.88	42.79	42.48	0.9	0.7
16	Data forecast	0.3	37.26	34.01	33.85	9.2	-
17	Data forecast	0.6	29.71	27.11	27.32	8.0	-0.8
18	Data forecast	0.9	42.88	42.02	41.95	2.2	-

$$\Delta_1 = 100 \cdot (\text{MBS} - \text{DJAH}) / \text{MBS}$$

$$\Delta_2 = 100 \cdot (\text{MCR} - \text{DJAH}) / \text{MCR}$$

Table 7a. Averaged cost price per product for  $N = 1, M = 1$ .

No.	Factor	$\rho$	MBS		MCR		DJAH	
			w	b	w	b	w	b
1	Default	0.3	17.84	3.09	12.46	3.30	9.12	2.84
2	Default	0.6	15.49	4.22	10.91	4.12	10.39	3.92
3	Default	0.9	30.88	5.00	28.46	4.79	28.42	4.77
4	Uniform arrivals	0.3	16.69	3.00	13.85	2.77	12.11	2.58
5	Uniform arrivals	0.6	12.59	4.00	11.74	4.00	11.29	3.87
6	Uniform arrivals	0.9	11.93	5.00	11.36	4.86	11.43	4.88
7	Capacity	0.3	14.41	4.32	10.14	4.68	9.04	4.13
8	Capacity	0.6	12.96	6.51	10.63	7.29	10.46	6.78
9	Capacity	0.9	21.17	10.00	18.92	9.40	19.10	9.28
10	Service time	0.3	24.80	2.28	17.13	2.70	14.35	2.47
11	Service time	0.6	27.80	3.62	19.87	3.75	19.83	3.70
12	Service time	0.9	65.29	5.00	60.86	4.73	60.57	4.73
13	Data missing	0.3	17.84	3.09	16.64	2.99	10.38	2.49
14	Data missing	0.6	15.49	4.22	13.42	3.91	11.66	3.68
15	Data missing	0.9	30.88	5.00	30.13	4.74	29.71	4.70
16	Data forecast	0.3	17.84	3.09	15.72	3.28	12.03	2.75
17	Data forecast	0.6	15.49	4.22	12.44	4.09	11.86	3.88
18	Data forecast	0.9	30.88	5.00	29.47	4.78	29.34	4.76

w = average waiting time

b = average batch size

Table 7b. Average waiting time and average batch size for  $N = 1, M = 1$

It can be concluded from Table 7a that operating costs for the various system configurations may show a significant difference. In the example situation, the choice for a machine with a higher capacity may imply lower cost price (settings 2 and 7).

The DJAH heuristic was compared with the best MBS strategy and the adapted version of the MCR heuristic. As can be concluded from a comparison of Tables 3 and 7a, relative differences between DJAH and MBS are smaller in the case where a minimal cost criterion is adopted than if the minimization of average flow time is taken as a criterion. For most settings marginal differences are found between MCR and DJAH. For higher oven capacities MCR has a minor advantage over DJAH. On the other hand DJAH proves to be more robust in case of missing data. These findings are consistent with the observations for the situation where the minimization of average flow time was taken as a criterion for optimization. However, it should be noticed that DJAH yields lower values for the average waiting time (Table 7b). From a viewpoint of customer service this may give DJAH an important advantage.

#### 4.2.2. *Multiple products, single machine*

For multiple products situations the inefficiency of the MCR heuristic, which was already observed for the single product case, becomes even more apparent, as is shown by the results in Tables 8a, b. Especially for product dependent service times, MCR performs poorly; for low and high traffic intensities even MBS shows better results.

It turns out that, in contrast with the spirit of a shortest processing time rule, MCR chooses small batch sizes for products that require a long processing time and large batch sizes for products which require a short processing time. As may be expected, high waiting costs are incurred by this choice, which are no longer compensated by low setup costs.

As can be concluded from the above tables (Tables 8a, b), DJAH performs better for all settings studied.

#### 4.2.3. *Single product, multiple machines*

To evaluate the single product multiple machines case, DJAH was compared with the best MBS strategy (Table 9). Again it was found that the minimum batch size should be chosen carefully. A wrong choice may lead to high operating costs. The relative differences found between MBS and DJAH are significantly smaller than those found for the same case with the average flow time criterion. This is consistent with the findings for the single product single machine case.

It is interesting to compare the results for both robustness tests (setting 13–18). For the settings in which the decision maker is confronted with an incomplete data set, it is found that MBS shows better performance than DJAH. Although differences are small, especially in view of the percentage of missing data (50%), the outcomes show that the error introduced as a consequence of missing data is no longer compensated by a better quality of decisions based upon information on future arrivals.

It should be noticed that, though cost prices are used here to estimate relative performance of the different control strategies, the same type of information can be used to support investment analysis. For example, from an operating cost point of view, a system which consists of two ovens, each having a capacity of 5, is preferable over a system which consists of one oven, that has a capacity of 10 (compare Tables

No.	Factor	$\rho$	MBSX	MCR	DJAH	$\Delta_1$	$\Delta_2$
1	Default	0.3	54.51	43.96	41.64	23.6	5.3
2	Default	0.6	43.42	36.79	36.85	15.1	—
3	Default	0.9	56.33	52.48	52.57	6.7	—
4	Uniform arrivals	0.3	59.33	45.04	44.33	25.3	1.6
5	Uniform arrivals	0.6	42.73	36.88	36.80	13.9	0.2
6	Uniform arrivals	0.9	40.73	37.58	37.60	7.7	—
7	Product mix	0.3	53.02	43.04	39.76	25.0	7.6
8	Product mix	0.6	42.55	36.08	35.76	16.0	0.9
9	Product mix	0.9	55.91	51.83	51.91	7.2	—
10	Capacities	0.3	57.78	48.23	43.98	23.9	8.8
11	Capacities	0.6	49.53	42.05	41.59	16.0	1.1
12	Capacities	0.9	73.02	62.10	61.37	16.0	1.2
13	Service times	0.3	55.73	56.71	41.82	25.0	26.3
14	Service times	0.6	46.40	46.00	37.76	18.6	17.9
15	Service times	0.9	68.66	88.84	55.35	19.4	37.7
16	Data missing	0.3	54.51	53.32	46.39	14.9	13.0
17	Data missing	0.6	43.42	40.98	39.33	9.4	4.1
18	Data missing	0.9	56.33	54.52	54.07	4.0	0.8
19	Data forecast	0.3	54.51	48.00	45.52	16.5	5.2
20	Data forecast	0.6	43.42	38.90	38.83	10.6	—
21	Data forecast	0.9	56.33	53.60	53.54	5.0	—

$$\Delta_1 = 100 * (\text{MBSX} - \text{DJAH}) / \text{MBSX}$$

$$\Delta_2 = 100 * (\text{MCR} - \text{DJAH}) / \text{MCR}$$

Table 8a. Averaged cost price per product for  $N = 2, M = 1$ .

No.	MBSX				MCR				DJAH			
	$w_1$	$w_2$	$b_1$	$b_2$	$w_1$	$w_2$	$b_1$	$b_2$	$w_1$	$w_2$	$b_1$	$b_2$
1	17.86	18.21	1.65	1.64	21.02	21.54	2.65	2.64	14.40	14.71	2.22	2.21
2	23.44	23.79	3.03	3.03	20.24	20.38	3.64	3.64	19.46	19.80	3.49	3.48
3	42.93	43.09	4.51	4.50	40.08	39.24	4.68	4.68	38.55	40.84	4.66	4.66
4	19.27	19.38	1.50	1.50	20.11	20.39	2.42	2.42	15.65	15.59	2.10	2.08
5	22.72	22.74	3.02	2.98	19.83	20.03	3.54	3.54	19.35	19.36	3.45	3.43
6	27.40	27.42	4.51	4.50	24.81	24.79	4.70	4.69	24.79	24.83	4.69	4.69
7	14.04	24.85	1.75	1.42	14.63	40.73	2.93	2.30	11.39	20.29	2.59	1.71
8	17.95	37.02	3.21	2.58	14.31	36.85	3.94	3.18	14.53	31.41	3.83	2.85
9	32.46	72.96	4.60	4.24	30.14	65.72	4.77	4.46	30.75	63.88	4.77	4.36
10	16.41	16.95	1.48	1.44	23.20	24.23	2.53	2.37	13.83	14.40	2.05	1.97
11	23.85	28.29	2.75	2.39	21.88	23.54	3.56	2.75	21.36	21.24	3.33	2.66
12	48.24	66.08	5.30	2.94	38.73	54.79	5.73	2.97	38.84	53.03	5.66	2.96
13	19.05	21.40	1.69	1.69	12.88	59.34	2.24	4.16	15.46	15.14	2.40	2.14
14	23.90	30.45	3.01	3.24	17.30	44.57	3.44	4.73	20.68	19.74	3.74	3.15
15	50.74	60.23	4.49	4.62	31.04	121.53	4.64	4.93	51.81	32.67	4.78	4.39
16	17.86	18.21	1.65	1.64	30.47	31.31	2.68	2.67	16.83	17.12	2.05	2.03
17	23.44	23.79	3.03	3.03	24.24	24.58	3.62	3.62	21.30	21.48	3.35	3.34
18	42.93	43.09	4.51	4.50	41.42	41.87	4.66	4.66	40.52	41.58	4.61	4.61
19	17.86	18.21	1.65	1.64	25.26	25.54	2.67	2.64	18.05	18.07	2.20	2.17
20	23.44	23.79	3.03	3.03	22.34	22.53	3.66	3.63	21.52	21.56	3.47	3.47
21	42.93	43.09	4.51	4.50	38.91	42.65	4.68	4.68	38.82	42.45	4.66	4.64

$w_1$  = average waiting time for products of type j

$b_1$  = average batch size for products of type j

Table 8b. Average waiting time and average batch size for  $N = 2, M = 1$ .

No.	Factor	MBS				DJAH				$\Delta_1$
		$\rho$	w	b	p	w	b	p		
1	Default	0.3	12.54	4.00	27.54	6.61	3.50	23.75	13.8	
2	Default	0.6	8.66	5.00	20.66	5.93	4.45	19.42	6.1	
3	Default	0.9	14.21	5.00	26.21	13.64	4.93	25.82	1.5	
4	Uniform arrivals	0.3	12.52	4.00	27.52	9.07	3.30	27.25	1.0	
5	Uniform arrivals	0.6	8.34	5.00	20.34	7.31	4.64	20.24	0.5	
6	Uniform arrivals	0.9	5.80	5.00	17.80	5.80	5.00	17.80	—	
7	Capacity	0.3	10.43	6.01	20.41	5.54	4.90	17.79	12.9	
8	Capacity	0.6	7.63	8.14	15.00	5.49	7.62	13.36	10.9	
9	Capacity	0.9	9.93	10.00	15.93	9.14	9.62	15.38	3.5	
10	Service time	0.3	17.17	3.04	36.91	9.52	2.91	30.12	18.3	
11	Service time	0.6	14.71	4.15	29.17	10.22	4.08	24.91	14.5	
12	Service time	0.9	28.73	5.00	40.73	27.07	4.85	39.44	3.2	
13	Data missing	0.3	12.54	4.00	27.54	6.30	2.94	26.72	3.0	
14	Data missing	0.6	8.66	5.00	20.66	5.93	3.97	21.04	−1.8	
15	Data missing	0.9	14.21	5.00	26.21	13.96	4.79	26.48	−1.1	
16	Data forecast	0.3	12.54	4.00	27.54	7.63	3.35	25.56	7.3	
17	Data forecast	0.6	8.66	5.00	20.66	6.29	4.37	20.02	3.1	
18	Data forecast	0.9	14.21	5.00	26.21	13.80	4.91	26.01	0.7	
19	Machines av.	0.3	6.25	4.00	21.25	4.80	4.21	19.04	10.4	
20	Machines av.	0.6	4.20	5.00	16.20	3.39	4.77	15.96	1.4	
21	Machines av.	0.9	6.45	5.00	18.45	6.31	4.97	18.38	0.4	

$\Delta_1 = 100 \cdot (\text{MBS} - \text{DJAH}) / \text{MBS}$

w = average waiting time

b = average batch size

$\rho$  = average cost price

Table 9. Average cost price per product for  $N = 1, M = 2; 4$ .

7a and 9). One can check whether higher investment costs (interest, depreciation) overrule these operational costs or not.

#### 4.2.4. Multiple product, multiple machines

The last series of simulations concern a case with two machines and four types of products. The results presented in Table 10, are to a large extent consistent with the results presented in Table 6, where the same example was evaluated according to the average flow criterion. Again it is apparent that relative differences between DJAH and MBS are smaller for the minimal cost criterion.

### 5. Concluding remarks

Especially for complex systems, DJAH shows its strength as a control strategy. DJAH shows a flexible response to changes in system conditions, in contradiction to e.g. MBS whose performance depends on the right setting of the minimum batch size. Above that, DJAH can easily be adapted for different system configurations, different product specifications or even different cost structures.

To give some indication of relative performance and robustness of the different control strategies we summarize results in Figure 5. The figure consists of two diagrams. The left diagram concerns simulation results for settings in which average waiting time is taken as a criterion for optimization. In a similar way the right

No.	Factor	$\rho$	MBSX	DJAH	$\Delta_1$
1	Default	0.3	53.43	41.81	21.7
2	Default	0.6	40.92	35.62	13.0
3	Default	0.9	45.33	42.73	5.7
4	Uniform arrivals	0.3	55.75	42.91	23.0
5	Uniform arrivals	0.6	41.16	35.47	13.8
6	Uniform arrivals	0.9	39.88	36.40	8.7
7	Product mix	0.3	51.17	39.64	22.5
8	Product mix	0.6	39.38	34.07	13.5
9	Product mix	0.9	44.28	41.78	5.6
10	Capacities	0.3	59.57	48.15	19.2
11	Capacities	0.6	49.37	41.03	16.9
12	Capacities	0.9	65.33	51.36	21.4
13	Service times	0.3	53.86	41.76	22.5
14	Service times	0.6	41.41	35.78	13.6
15	Service times	0.9	47.40	43.54	8.1
16	Data missing	0.3	53.43	44.91	15.9
17	Data missing	0.6	40.92	37.34	8.7
18	Data missing	0.9	45.33	43.20	4.7
19	Data forecast	0.3	53.43	43.74	18.1
20	Data forecast	0.6	40.92	36.63	10.5
21	Data forecast	0.9	45.33	43.13	4.9

$$\Delta_1 = 100 \cdot (\text{MBSX} - \text{DJAH}) / \text{MBSX}$$

Table 10. Average cost price per product for N = 4, M = 2.

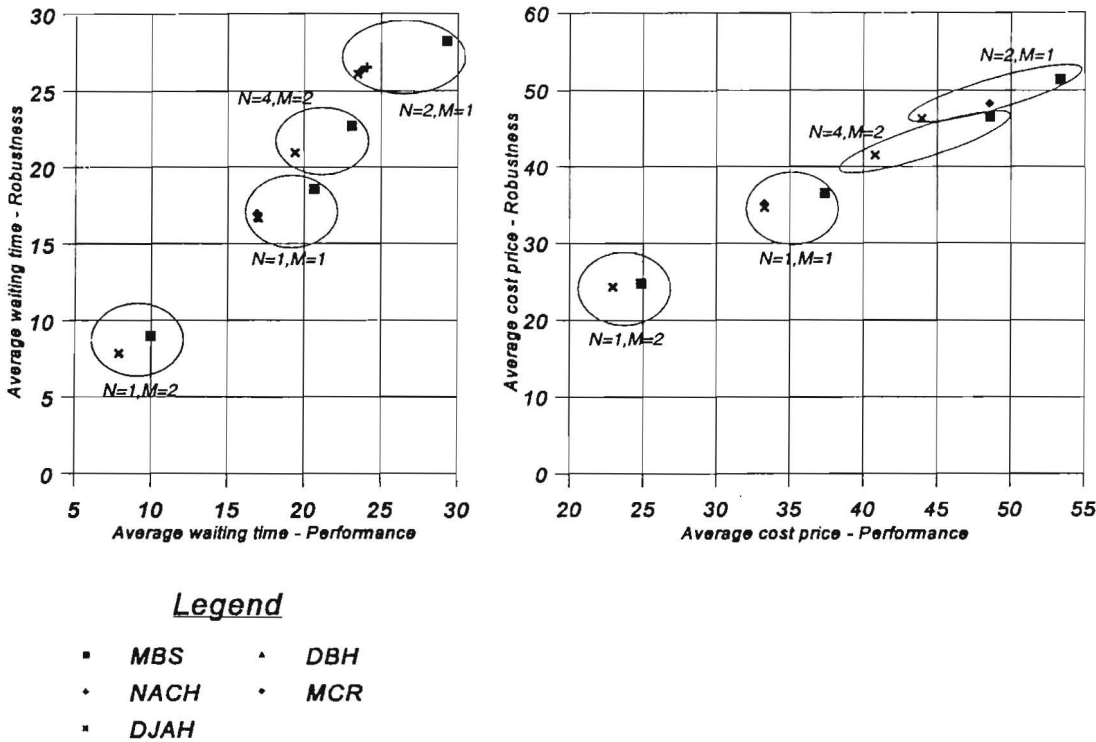


Figure 5. Performance and robustness of control strategies for batch processing systems.

diagram presents simulation results for situations in which average cost price per product is taken as a criterion for optimization. Results are clustered according to a specific setting for the number of product types ( $N$ ) and the number of machines ( $M$ ), e.g.  $N = 1$ ,  $M = 2$ . Performance is computed as the average of simulation results presented in this paper for situations in which arrivals are known with certainty. Robustness is related to the response of look-ahead strategies to situations where future arrivals are forecast or data on future arrivals are incomplete. It can be seen as the performance under uncertainty. Robustness is computed as the average of simulation results – concerning average waiting time and average cost price – for the corresponding case-situations mentioned in this paper. Note that this way of presenting simulation results may conceal some of the potential strengths and weaknesses of the different control strategies for specific settings. Nevertheless the figures are helpful in getting an overview of the differences between the distinguished control strategies with regard to performance and robustness. Conclusions which may be drawn from the figure are:

- System performance is considerably improved if information on future arrivals is included in decision making (compare results for MBS and the look-ahead strategies)
- Simulation results for the different look-ahead strategies differ by a small margin for most settings. In case  $N > 1$  DJAH shows better performance/robustness than other look-ahead strategies.

Because DJAH only considers identical machines, future research will be directed at bulk queueing systems which consist of machines that have different characteristics as far as, for example, capacity and or processing time is concerned. These differences may refer to product specifications, which for instance put demands on the size of a machine, but also they may refer to underlying cost structures. Also new fields of application for look-ahead strategies will be explored, for example transportation systems.

## References

- BAGCHI, T. P., and TEMPLETON, J. G. C., 1972, Numerical methods in Markov chains and bulk queues. In *Lecture Notes in Economics and Mathematical Systems*, M. Beckmann, G. Goos and H. P. Künzi (eds), 72, (Berlin: Springer).
- BAKKENIST MANAGEMENT CONSULTANTS, 1994, ExSpect User Manual (Amsterdam: Bakkenist ExSpect).
- BLACKSTONE, J. H., PHILLIPS, D. T., and HOGG, G. L., 1982, A state-of-the-art survey of dispatching rules for manufacturing job shop operations. *International Journal of Production Research*, 20 (1), 27–45.
- BOOCH, G., 1994, *Object-Oriented Analysis and Design—With Applications* (Redwood City: Benjamin Cummings).
- CHAUDRY, M. L., and TEMPLETON, J. G. C., 1983, *A First Course in Bulk Queues* (New York: Wiley).
- CONWAY, R. W., MAXWELL, W. L., and MILLER, L. W., 1967, *Theory of Scheduling* (Reading, MA: Addison Wesley).
- DEB, R. K., and SERFOZO, R. F., 1973, Optimal control of batch service queues. *Advances in Applied Probability*, 5, 340–361.
- FOWLER, J. W., HOGG, G. L., and PHILLIPS, D. T., 1992, Control of multiproduct bulk service diffusion/oxidation processes. *IIE Transactions*, 24 (4), 84–96.
- GIBBONS, J. D., and CHAKRABORTI, S., 1992, Nonparametric statistical inference. In *Statistics: Textbooks and Monographs*, D. B. Owen and W. R. Schucany (eds) 131 (New York: Marcel Dekker).

- GLASSEY, C. R., and WENG, W. W., 1991, Dynamic batching heuristic for simultaneous processing. *IEEE Transactions on Semiconductor Manufacturing*, **4** (2), 77–82.
- GLASSEY, C. R., MARKGRAF, F., and FROMM, H., 1993, Real time scheduling of batch operations. In *Optimization in Industry: Mathematical Programming and Modeling Techniques in Practice*, T. A. Ciriani and R. C. Leachman (eds) (Chichester: Wiley), pp. 113–137.
- HODES, B., SCHOONHOVEN, B., and SWART, R., 1992, On line planning van ovens. Technical report, University of Twente, The Netherlands (in Dutch).
- HOOVER, S. V., and PERRY, R. F., 1989, *Simulation – A Problem Solving Approach* (Reading, MA: Addison Wesley).
- LAW, A. M., and KELTON, W. D., 1991, *Simulation Modeling and Analysis* (Singapore: McGraw-Hill).
- LITTLE, J. D. C., 1961, A proof for the queueing formula:  $L = \lambda W$ . *Operations Research*, **9**, 383–387.
- NEUTS, M. E., 1967, A general class of bulk queues with Poisson input. *Annals of Mathematical Statistics*, **38**, 759–770.
- ROBINSON, J. K., FOWLER, J. W., and BARD, J. F., 1995, The use of upstream and downstream information in scheduling semiconductor batch operations. *International Journal of Production Research*, **33** (8), 1849–1869.
- SILVER, E. A., and MEAL, H. C., 1973, A heuristic for selecting lot size quantities for the case of a deterministic time varying demand rate and discrete opportunities for replenishment. *Production and Inventory Management*, **14**, 64–74.
- UZSOY, R., LEE, C. Y., and MARTIN-VEGA, L. A., 1992, A review of production planning and scheduling models in the semiconductory industry, Part I: System characteristics, performance evaluation and production planning. *IIE Transactions*, **24** (4), 47–60.
- UZSOY, R., LEE, C. Y., and MARTIN-VEGA, L. A., 1994, A review of production planning and scheduling models in the semi-conductor industry, Part II: Shop-floor control. *IIE Transactions*, **26** (5), 44–55.
- WENG, W. W., and LEACHMAN, R. C., 1993, An improved methodology for real-time production decisions at batch-process workstations. *IEEE Transactions on Semiconductor Manufacturing*, **6** (3), 219–225.
- VAN DER ZEE, D. J., 1997, Simulation as a Tool for Logistics Management. PhD thesis, University of Twente, The Netherlands.
- VAN DER ZEE, D. J., HARTEN, A., and VAN SCHUUR, P. C., 1995, Business Simulation for Logistics Management, Technical report, University of Twente, The Netherlands.